

Lesson 2-6 Configurable UART

Welcome back to Cypress Academy, WICED WiFi 101. In the last video I used the built-in and pre-configured debug UART to display information in a terminal window. But, what if you want to use a different configuration – for example a different baud rate – or if you want to accept input from the terminal in addition to sending the output to the terminal? I will cover both of these cases in this video.

First, let's send data to the UART at 9600 baud instead of the 115200 baud from the previous example. I'll start by copying the 05_interrupt project to a new project called 10_uartsend and then update these files as necessary.

I am going to re-use the same physical UART pins as the one used for WPRINT_APP_INFO example, so I need to tell the SDK not to initialize it by default. That's done by adding the following line to the makefile:

```
GLOBAL_DEFINES += WICED_DISABLE_STDIO
```

You can use this same method to define any macro at build time. In this case, this line has exactly the same effect as doing “#define WICED_DISABLE_STDIO”

Now that I have disabled the default UART settings, I need to configure and initialize the UART with my desired settings. I start by setting up the configuration structure as shown below and then by calling the initialization function with that structure.

Notice that you can find the allowed selections for each parameter by using “Open Declaration” as I showed you in the earlier video.

Now that the UART is configured and initialized, I can send data using the wiced_uart_transmit_bytes function. In this case, I will print a single digit representing how many times the button has been pressed.

Now I will program the board and test it out using a putty serial terminal set for 9600 baud.

Next, let's create a project that will receive input from the terminal. I will use a press of the “0” on the keyboard to turn the LED off and a press of the “1” to turn it on.

First, I will copy 10_uartsend to a new project which I will fix and rename 11_uartreceive. And, as before, I will make all of the necessary updates to the project configuration and the makefile, etc. Note that the WICED_DISABLE_STDIO define is needed in the makefile for this project as well.

In this case, in the initialization function, I provide a receive buffer in addition to the UART configuration. This buffer is where the receive data will be stored. I read data

from the buffer using the `wiced_uart_receive_bytes` API and then turn the LED on or off depending on the character received.

Now, let's program the board and test it out. I can turn the LED on and off from the keyboard.

In the next video, I'll show you how to use the PWMs that are built into the chip.

You can post your comments and questions in our WiFi developer community, or, as always, you are welcome to email me at alan_hawse@cypress.com or tweet me at [@askiotexpert](https://twitter.com/askiotexpert). Thank you.